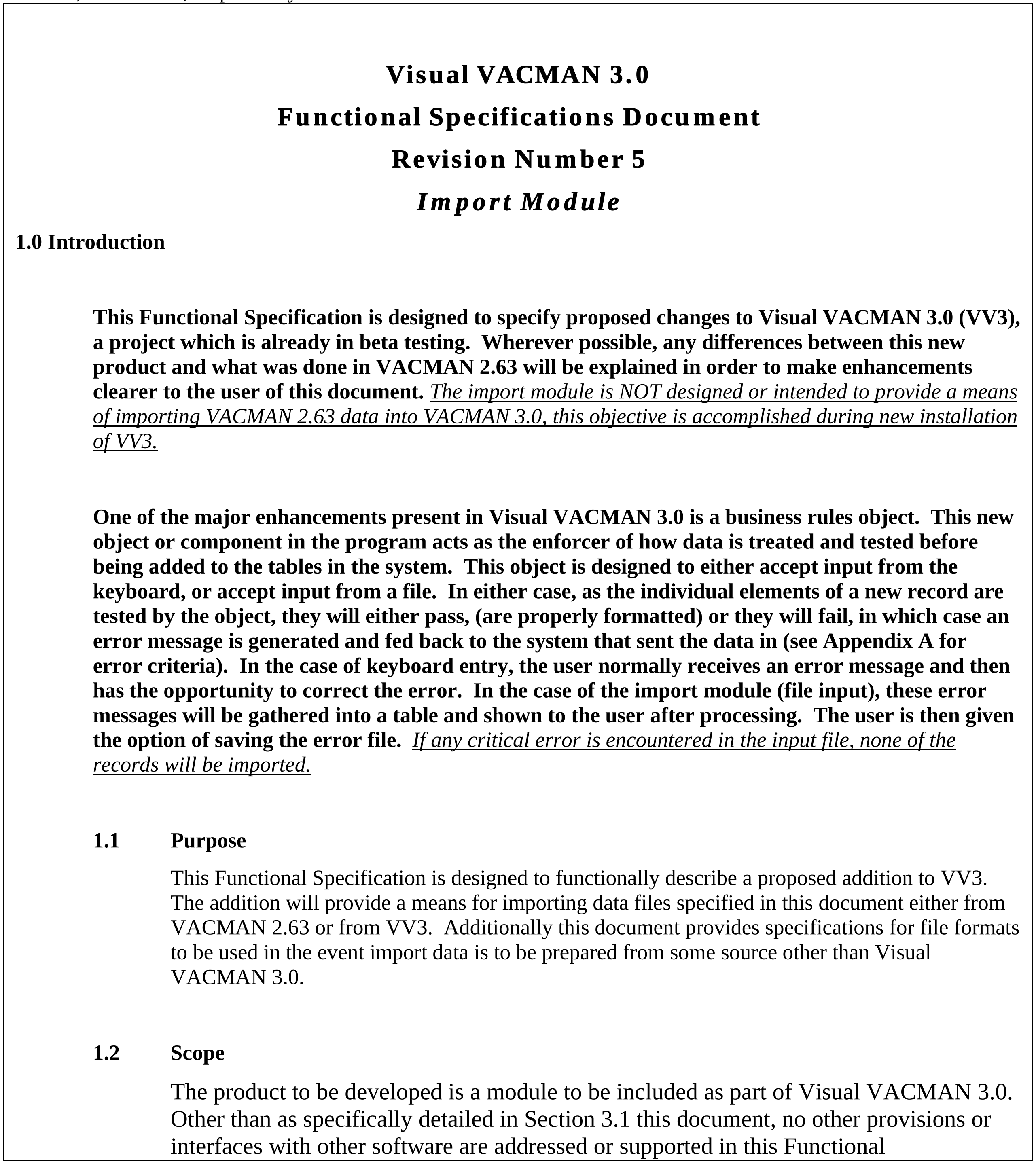
Appendix E – Import Functional Specifications

The following information provides specifications on the Import Function in VACMAN.

Important Notes!

- The import module is **NOT** designed or intended to provide a means of importing VACMAN data into a subsequent version of VACMAN. That objective is accomplished during new installation of VACMAN 3.
- If the target file type is of type Excel worksheet or Lotus worksheet, VACMAN will not copy more than 16,383 or 8,191 records, respectively.



Visual VACMAN 3.0

Functional Specifications Document

Revision Number 5

Import Module

1.0 Introduction

This Functional Specification is designed to specify proposed changes to Visual VACMAN 3.0 (VV3), a project which is already in beta testing. Wherever possible, any differences between this new product and what was done in VACMAN 2.63 will be explained in order to make enhancements clearer to the user of this document. *The import module is NOT designed or intended to provide a means of importing VACMAN 2.63 data into VACMAN 3.0, this objective is accomplished during new installation of VV3.*

One of the major enhancements present in Visual VACMAN 3.0 is a business rules object. This new object or component in the program acts as the enforcer of how data is treated and tested before being added to the tables in the system. This object is designed to either accept input from the keyboard, or accept input from a file. In either case, as the individual elements of a new record are tested by the object, they will either pass, (are properly formatted) or they will fail, in which case an error message is generated and fed back to the system that sent the data in (see Appendix A for error criteria). In the case of keyboard entry, the user normally receives an error message and then has the opportunity to correct the error. In the case of the import module (file input), these error messages will be gathered into a table and shown to the user after processing. The user is then given the option of saving the error file. *If any critical error is encountered in the input file, none of the records will be imported.*

1.1 Purpose

This Functional Specification is designed to functionally describe a proposed addition to VV3. The addition will provide a means for importing data files specified in this document either from VACMAN 2.63 or from VV3. Additionally this document provides specifications for file formats to be used in the event import data is to be prepared from some source other than Visual VACMAN 3.0.

1.2 Scope

The product to be developed is a module to be included as part of Visual VACMAN 3.0. Other than as specifically detailed in Section 3.1 this document, no other provisions or interfaces with other software are addressed or supported in this Functional

Specification.

1.3 Definitions, acronyms, and abbreviations

VV3 – Visual VACMAN 3.0, the product under development and beta testing.

1.4 References

1.4.1 VACMAN 2.63 Import Module Requirements Document.

1.4.2 Appendix A – Error Checking Criteria

1.5 Overview

The remaining portions of this document are organized to specify the end user required skills, overall functional changes to the existing product, hardware and network specific requirements (if any) and any other requirements specific to this product.

2.0 Overall Description

2.1 Product perspective

This Functional Specification is being generated to document proposed addition to an already existing product, currently in the final stages of Beta Testing. While the changes to be made are functional in nature, the overall impact on the existing product is minimal. The current flow of data and reports of this document remain unchanged.

2.2 Product Functions

The implementation will include:

- Provide a means of validating and inputting records from standard tables (.dbf files)
- Provide a means of validating and inputting records from standard Excel spreadsheet files (.xls files)
- Provides a means of validating and inputting records from standard lotus spreadsheet files (.wk1 files)
- Provide a means of validating and inputting records from standard comma delimited text files (.txt files)
- Provide a means of validating and inputting records from fixed length data files (.sdf files)
- Data formatted in any of the above listed files will be used to update the following information.
  - Distribution data (normal project distributions)
  - Distribution data from other than project transactions data
  - Varicella order data
  - Provider information
  - Enrollment information
  - Physician information

- Provider inventory data
- Doses administered information
- New varicella orders

2.3 User Characteristics

The end user of this product will have the same basic skill set as current VACMAN 2.63 users. Users should have basic computer skills including the following:

- Able to successfully start their machine and log into their local area network.
- Able to input data on typical data input screens.
- Basic keyboarding skills.
- Basic mouse and other input device operation.
- In addition, the user should be aware of when and how data is to be imported into VACMAN and any potential dangers to data integrity posed by improper use of the module.
- User must have security level 2 or 3 to use this utility.

2.4 General Constraints

2.4.1 Equipment – Equipment required to support this product enhancement are identical to those already specified for Visual VACMAN 3.0 as a whole. No additional requirements are specified.

2.4.2 Required software – Software required to support this product enhancement are identical to those already specified for Visual VACMAN 3.0 as a whole. No additional requirements are specified.

2.4.3 Memory available - Memory requirements for this product enhancement are identical to those already specified for Visual VACMAN 3.0 as a whole. No additional requirements are specified.

2.4.4 Operating system – Operating System requirements to support this product enhancement are identical to those already specified for Visual VACMAN 3.0 as a whole. No additional requirements are specified.

2.4.5 Network configuration – Network configuration required to support this product enhancement are identical to those already specified for Visual VACMAN 3.0 as a whole. No additional requirements are specified.

2.4.6 Other constraints on the system – Users must have the ability to access Visual VACMAN 3.0.

2.5 Assumptions and Dependencies

This module is designed to be part of Visual VACMAN 3.0 as a whole. It will not be accessible as a stand alone process nor will it be designed to be accessed from outside the CDC network. The product will not be designed to provide information to users outside the currently existing network.

a.

3.0 Specific requirements

The following Functional requirements will be the basis for all requirements specified for this product enhancement. There are no other requirements for this enhancement other than as specifically described in the functional requirements listed.

3.1 Functional Requirements

3.1.1 User Interface Screen – The user will be permitted to select import options from an interface screen similar to the one provided in VACMAN 2.63. The screen will give the user the option to choose from the following file import types:

- Standard database files (.dbf files)
- Fixed length text files (.sdf files or space delimited files)
- Comma delimited text files (.txt files)
- Microsoft Excel spreadsheet files (.xls files)
- Lotus spreadsheet files (.wk1 files)

The user will also be permitted to select which data set to be updated:

- Distribution data (from normal project sources)
- Distribution data (from non-project sources)
- Varicella vaccine ordering data
- Provider information
- Provider inventory information
- Provider enrollment information
- Doses administered information
- Physician information
- New varicella orders

The user will be provided with a browse button or similar interface to allow them to select the appropriate file from which to import the data set.

Note: In VM 2.63 the user was also permitted to choose between condensed and full file format. This feature is not necessary in this module as the file types mentioned are already supported by the file definitions mentioned above.

3.1.2 Distribution data from Project Depots

3.1.2.1 Import file formats:

3.1.2.1.1 Database Files:

Data to be imported in the form of a table will be formatted with the following field definitions:

Source Table in VACMAN: distrib.dbf

Field Name	**Type**	**Size**
input	Date	8
xmitdate	Date	8
dateout	Date	8
pin	Char	6
frompin	Char	6
action	Numeric	2,0
lot	Char	15
ordsize	Numeric	2,0
vaccine	Char	8
units	Numeric	8,0
doses	Numeric	8,0
fdose1	Numeric	8,0
fdose2	Numeric	8,0
fdose3	Numeric	8,0
seekid	Numeric	9,0
cdcpo	Numeric	10,0
status	Char	2
shipdate	Date	8
orddose	Numeric	8,0
shipped	Numeric	1,0
notes	Char	25
chgdate	Date	8
userid	Char	8
distid	Integer	4

Note: In Visual VACMAN 3.0 the distid field is the primary key for the distrib table and must not be altered in any way as this will prevent the record from matching in the distrib table and could cause problems with calculation of lot quantities on hand.

3.1.2.1.2 Comma Delimited Files:

Comma delimited files will have a file extension of .txt. Data fields will be exported from the distrib table in VV3 or from VACMAN 2.63 into an ASCII standard text file containing one line of text for each record in the table.

- Each field will be separated by a comma and no extra spaces.
- Fields containing char data will be surrounded by double quotes.
- Fields containing numeric data will be trimmed to their minimum length.
- Fields containing date data will be formatted to 8 spaces containing YYYYMMDD data. In the event there is no data in the corresponding table field, the appropriate space in the text file will contain 8 spaces.
- Data will be arranged in the same order as listed as section 3.1.2.1.1 above.
- Files of this type can be most easily created by using the following commands from the FoxPro command interface:
    - Use distrib
    - Copy to XXXXX.txt delimited (substitute file name for XXXXX)
    - **Note**: formatting of the file generated from FoxPro 2.6 for Screens and the one formatted by Visual FoxPro 7.0 (VFP7) will vary slightly. Either file can be successfully imported by Visual VACMAN 3.0.

3.1.2.1.3 Space Delimited Files:

Space Delimited Files will be formatted in a manner similar to specified for Comma Delimited text files, however space delimited files contain a single line of data for each record in the table with no delimiting character. SDF files format the data in streams of values of exactly the same length as the corresponding fields in the table from which the data is exported. No quotation marks or other special characters for dates are required to distinguish data types. The data must be arranged in the same order and to the same lengths as specified in section 3.1.2.1.1 above.

A space delimited file may be most easily created by issuing the following commands from the command line interface in FoxPro:

- Use distrib
- Copy to filename.sdf SDF (substitute filename to be created).

3.1.2.1.4 Microsoft Excel Spreadsheet files (.xls files)

Excel spreadsheet files will be formatted with a column heading corresponding to the name of each field as defined in section 3.1.2.1.1 above. Each record to be imported will be listed on a single row of the spreadsheet in the same order and with the same formatting as defined in section 3.1.2.1.1. An Excel spreadsheet file may be most easily created by issuing the following commands from the command line interface of FoxPro:

- Use distrib
- Copy to filename.xls XLS (substitute filename to be created)

3.1.2.1.5 Lotus Spreadsheet files (.wk1 files)

Lotus spreadsheet files will be formatted in exactly the same manner as described in section 3.1.2.1.4 above. A Lotus spreadsheet file may be most easily created by issuing the following commands from the command line interface in FoxPro:

- Use distrib
- Copy to filename.wk1 WK1 (substitute filename to be created)

3.1.2.1.5 Other file formats

File formats other than those specified above will not be supported by this module. Regardless of how the file is created, the formatting described in section 3.1.2.1 of this document must be followed. If the file to be imported fails to meet acceptable criteria it will be rejected with the appropriate error message to the user. Best results for file creation will be obtained by using the file export utility in VV3.0.

3.1.2.2 Data Import:

- After selection of the appropriate file and file type for distribution data from project depots from the user interface screen described in section 3.1.1 above, the system will copy the new data into an import table meeting the same table definition as the distrib.dbf file currently in use in Visual VACMAN 3.0.
- The user will then be provided with an interface screen prompting the user for the type of input to be conducted. After error checking import functions will be carried out as follows:
    - **Import Archived Records** – This option will match all of the records in the import file and will only append records that do not exist in the distrib table at the time of the action.
        - This action is the mirror to the export function that allows for the removal of distribution records currently in place. The export program copies data into the disthist table when moving records to archive.
        - **Note:** In order to properly calculate distributions, the import program must move the archived data out of the disthist table and then make a matching entry in the distrib table. If an attempt is made to input data using the ARCHIVE function of the import system and the data file used was not created by the export utility, recalculation of the distrib table, lots and other data tables may be corrupted. **Data created outside of VACMAN should NOT be used for the archive function…**
        - **Note:** If any duplicate records are found in the import file not matching records in the distrib file, they will be ignored. The import function relies on an exact match of the distid field, if this has been tampered with, the records will not match. The user will be prompted with the appropriate error message.
        - Archived data should not be tampered with in any way. If the user wishes to enter data that has been altered the update or add functions should be used. **Other than checking for record matches and recalculation of lots balances, NO ERROR CHECKING IS DONE.**
    - **Update records** – Only those records that currently exist in the distrib table will be imported.
        - Records that do not already exist will be moved to a separate file and an error record will be saved to an error file.
        - As the records are matched and updated into the distrib table in Visual VACMAN 3.0, the new information will be treated by the business rules object as an edit.
        - **Note:** The seekid field in **ALL** of the records in the import file

must match before **ANY** of the records will be imported, this standard was also present in VACMAN 2.63. If all of the records do not match, the user will be presented with an appropriate message and given the chance to view and save the errors found. Additionally the user will have the opportunity to view the records that did not pass duplicate checking and save them to a separate file.

- **Note:** If the end user intends to make changes to data external to Visual VACMAN 3.0, the seekid key field, along with action, dateout, fromPIN, PIN, vaccine, and lot must not be changed. If they are changed the record cannot be matched to a current record in the distrib table, and the system will not allow the record to be imported.
- Changes to lot inventories and other calculated data will be conducted in accordance with current business rules.

- **Add new records** - in this option, only records that are NOT duplicates will be processed.
  - New records will be handled in accordance with the same business rules that would be used for a standard keyboard entry.
  - **Note:** in version 2.63, the user had the opportunity to select an option that would check whether excess ordering has been conducted. This function is now a part of the business rules and will always be conducted.
  - When the user clicks on the add option on the main user interface screen two additional options will be made available: the user can auto-assign a shipment date when the shipdate field in the import table is empty and auto-assign lot numbers when they are left empty. A radio box will be provided for each option and a separate field provided that the user may use to input a desired ship date. By default the date will be the current day.
  - **Note:** distid and seekid (unique fields) will be overwritten if present by system generated identifiers in add mode. If importing data using the add mode, the values originally present in these fields will be changed. The import file itself is not changed, only the values entered in the distrib table.

3.1.2.2.4 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.3 Distribution data (Other Transactions)

3.1.3.1 Import file formats:

3.1.3.1.1 Database Files:

Source Table in VACMAN: Distrib.dbf

Table definitions for this portion of the import module are identical to those described in section 3.1.2.1.1.

3.1.3.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.3.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.3.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.3.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.3.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.3.2 Data Import procedures:

Import procedures for this portion of the import module are almost identical to that described in section 3.1.2.2. In the 2.63 version of the product, there were no options for handling archive, update and add records, the system handled all records as an update. The new module will provide a choice with the same parameters as described in section 3.1.2.2 above.

3.1.3.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.4 Varicella Data

3.1.4.1 Import file formats:

3.1.4.1.1 Database Files:

Source Table in VACMAN: stock.dbf

Field Name	Type	Size
rectype	Char	1
pin	Char	6
orddate	Char	12
ordNum	Char	10
orddose	Numeric	8,0
xmitdate	Char	12
datein	Char	12
rcvxmit	Char	12
timein	Char	5
signee	Char	25
vaccine	Char	8
units	Numeric	8,0
size	Numeric	4,0
doses	Numeric	8,0
maker	Char	8
lot	Char	15
expiry	Char	12
cost	Numeric	9,4
contract	Numeric	15,0
fcode1	Char	1
fdose1	Numeric	8,0
fcode2	Char	1
fdose2	Numeric	8,0
fcode3	Char	1
fdose3	Numeric	8,0
uzr	Char	10
remain	Numeric	8,0
seekid	Numeric	9,0
cdcpo	Numeric	10,0
podate	Char	12
nipord	Numeric	8,0
status	Char	2
note	Char	50
chgdate	Char	12
statepo	Char	32
userid	Char	8

rcvchgdt	Char	12
ordpin	Char	6
ordsize	Numeric	4,0
ndcno	Char	16
exctax	Numeric	9,4
addtoinv	Numeric	1,0
stockid	Integer	4

Note: In Visual VACMAN 3.0 the stockid field is the primary key for the stock table and must not be altered in any way as this may prevent the record from matching.

3.1.4.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.4.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.4.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.4.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.4.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.4.2 Data Import procedures:

- The user will select Varicella Order Data, the appropriate file type to import, browse to the file name on the interface screen, select the import type (add, update, or archive) and click the continue button.
- **Note:** Excess order checking – an option in VACMAN 2.63 is automatically checked in Visual VACMAN 3.0 – the option to check or not check this value will no longer be available in this module.
- The system will check the import file for duplicate records, and varicella orders which have already been transmitted (the record in the import file matches a record in the stock table based on a match of PIN, order date, vaccine and fundcode1). Alternatively, if the stockid field is present in the import file, this value will be used

to check for a duplicate. Records which are duplicates or which have already been transmitted will not be imported.

- If the number of records left is less than 1, the user is prompted with an appropriate notice that there are no records to import.
- Records which are ready to be imported will be checked for errors as discussed in section 3.1.4.2.1.
- On successful completion of error checking the system will scan through the import file matching against the stock table looking for matches of order date, vaccine, and fundcode1 values. If a match is found and there is no transmit date, the import record will overwrite the existing record in stock. If there is no match, the new record will be added to the stock table or in the update mode the record will be updated.
- For archival data, the functionality is similar to that described for distribution records above.
- **Note:** This import mechanism is not intended to permit the uploading of NEW varicella orders. New varicella orders are handled as described in section 3.1.10 of this specification. The import function described in section 3.1.4 (this section) is only intended to update varicella orders in the stock table.

3.1.4.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.5 Provider Data

3.1.5.1 Import file formats:

3.1.5.1.1 Database Files:

Source Table in VACMAN: Provider.dbf

Field Name	**Type**	**Size**
pin	Char	6
linkpin	Char	6
alphaid	Char	15
firstname	Char	15
initial	Char	1
lastname	Char	35
title	Char	9
contact	Char	30
address	Char	35
addres2	Char	35
city	Char	20

state	Char	2
zipcode	Char	10
ml_contact	Char	30
ml_address	Char	35
ml_addres2	Char	35
ml_city	Char	20
ml_state	Char	2
ml_zipcode	Char	10
county	Char	15
phone	Char	14
extension	Char	4
fax	Char	14
modem	Char	14
subdiv	Char	15
sector	Numeric	2,0
cat1	Char	10
cat2	Char	10
cat3	Char	10
note	Char	50
active	Char	1
codes	Char	10
physcount	Numeric	2,0
origdate	Date	8
other	Char	10
datevfc	Date	8
createdate	Date	8
xmitdate	Date	8
chgdate	Date	8
userid	Char	8
phone2	Char	14
email	Char	40
cat4	Char	10
cat5	Char	10
provtype	Char	1
email2	Char	35
extension2	Char	4
fax2	Char	14

wh_note	Char	50
dateactive	Date	8
datexvfc	Date	8
provid	Integer	4

Note: In Visual VACMAN 3.0 the provid field is the primary key for the provider table and must not be altered in any way as this may prevent the record from matching.

3.1.5.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.5.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.5.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.5.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.5.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.5.2 Data Import procedures:

- ➢ The user will select Provider Data, the appropriate file type to import, the import option (archive, update or add), browse to the file name on the interface screen, and click the continue button.
- ➢ The system will check for duplicates by matching PIN number. Data matching and handling of duplicates will be similar to the rules described in section 3.1.2.2 with the exception that there is no calculation to be done.
- ➢ Choosing the Cancel button will abort the operation and close the import module.
- ➢ Regardless the choice of the type of import to do, error checking will first be conducted. If no critical errors are detected – the user will be required to view any warning messages and will then permitted to continue the import process as described previously.

3.1.5.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.6 Enrollment Data

3.1.6.1 Import file formats:

3.1.6.1.1 Database Files:

Source Table in VACMAN: enroll.dbf

Field Name	**Type**	**Size**
PIN	Char	6
AGEDEFINE	Char	6
ENSOURCE	Char	10
ALLKIDS1	Numeric	7,0
MEDICAID1	Numeric	7,0
UNINSURED1	Numeric	7,0
AMINDIAN1	Numeric	7,0
FQHC_RHC1	Numeric	7,0
VFCKIDS1	Numeric	7,0
G317KIDS1	Numeric	7,0
STATEKIDS1	Numeric	7,0
ALLKIDS2	Numeric	7,0
MEDICAID2	Numeric	7,0
UNINSURED2	Numeric	7,0
AMINDIAN2	Numeric	7,0
FQHC_RHC2	Numeric	7,0
VFCKIDS2	Numeric	7,0
G317KIDS2	Numeric	7,0
STATEKIDS2	Numeric	7,0
ALLKIDS3	Numeric	7,0
MEDICAID3	Numeric	7,0
UNINSURED3	Numeric	7,0
AMINDIAN3	Numeric	7,0
FQHC_RHC3	Numeric	7,0
VFCKIDS3	Numeric	7,0
G317KIDS3	Numeric	7,0

STATEKIDS3	Numeric	7,0
ALLKIDS4	Numeric	7,0
MEDICAID4	Numeric	7,0
UNINSURED4	Numeric	7,0
AMINDIAN4	Numeric	7,0
FQHC_RHC4	Numeric	7,0
VFCKIDS4	Numeric	7,0
G317KIDS4	Numeric	7,0
STATEKIDS4	Numeric	7,0
CHGDATE	Date	8
USERID	Char	8
OTHER1	Numeric	7,0
OTHER2	Numeric	7,0
OTHER3	Numeric	7,0
OTHER4	Numeric	7,0

Note: In Visual VACMAN 3.0 the pin field is the primary key for the enroll table and must not be altered in any way as this may prevent the record from matching.

3.1.6.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.6.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.6.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.6.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.6.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.6.2 Data Import procedures:

- The user will select Enrollment Data, the import option (archive, update or add), the appropriate file type to import, browse to the file name on the interface screen, and click the continue button.
- The system will check for duplicates by matching PIN number. Data matching and handling of duplicates will be similar to the rules described in section 3.1.2.2 with the exception that there is no calculation to be done.
- Choosing the Cancel button will abort the operation and close the import module.
- Regardless the choice of the type of import to do, error checking is first conducted. If no critical errors are detected – the user will be required to view any warning messages and then permitted to continue the import process as described previously.

3.1.6.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.7 Physician Data

3.1.7.1 Import file formats:

3.1.7.1.1 Database Files:

Source Table in VACMAN: physicn.dbf

Field Name	**Type**	**Size**
PIN	Char	6
PHYSID	Char	12
LASTNAME	Char	20
FIRSTNAME	Char	15
INITIAL	Char	1
TITLE	Char	9
OTHER1	Char	12
OTHER2	Char	12
MEDICAIDNO	Char	12
MEDLICENSE	Char	12
CHGDATE	Date	8
USERID	Char	8
ADDTOCOUNT	Numeric	1.0
DATEACTIVE	Date	8
PHYSNID	Integer	4

Note: In Visual VACMAN 3.0 the physnid field is the primary key for the physicn table and must not be altered in any way as this may prevent the record from matching.

3.1.7.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.7.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.7.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.7.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.7.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.7.2 Data Import procedures:

- The user will select Physician Data, the appropriate file type to import, the import option (archive, update or add), browse to the file name on the interface screen, and click the continue button.
- The system will check for duplicates by matching PIN number or physid if present. Data matching and handling of duplicates will be similar to the rules described in section 3.1.2.2 with the exception that there is no calculation to be done.
- Choosing the Cancel button will abort the operation and close the import module.
- Regardless the choice of the type of import to do, error checking will first be conducted. If no critical errors are detected – the user will be required to view any warning messages and will then permitted to continue the import process as described previously.

3.1.7.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.8 Doses Administered Data

3.1.8.1 Import file formats:

3.1.8.1.1 Database Files:

Source Table in VACMAN: dosesadm.dbf

Field Name	**Type**	**Size**
ADATE	Date	8
PIN	Char	6
SECTOR	Numeric	1,0
VACCINE	Char	8
POSITION	Numeric	2,0
DOSE	Numeric	1,0
G1	Numeric	7,0
G2	Numeric	7,0
G3	Numeric	7,0
G4	Numeric	7,0
G5	Numeric	7,0
G6	Numeric	7,0
G7	Numeric	7,0
G8	Numeric	7,0
G9	Numeric	7,0
G10	Numeric	7,0
G11	Numeric	7,0
G12	Numeric	7,0
G13	Numeric	7,0
ROWTOT	Numeric	10,0
DAUZR1	Numeric	7,0
DAUZR2	Numeric	7,0
DAUZR3	Numeric	7,0
CHGDATE	Date	8
USERID	Char	8
DAUZR4	Numeric	7.0
DAUZR5	Numeric	7.0
DAUZR6	Numeric	7.0
DAUZR7	Numeric	7.0
DAUZR8	Numeric	7.0

DAUZRTOT	Numeric	8.0
DOSESADMID	Integer	4

Note: In Visual VACMAN 3.0 the dosesadmid field is the primary key for the dosesadm table and must not be altered in any way as this may prevent the record from matching.

3.1.8.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.8.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.8.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.8.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.8.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.8.2 Data Import procedures:

- The user will select Doses Administered Data, the import option (archive, update or add), browse to the file name on the interface screen, and click the continue button.
- The system will check for duplicates by matching PIN number. Data matching and handling of duplicates will be similar to the rules described in section 3.1.2.2 with the exception that there is no calculation to be done.
- Choosing the Cancel button will abort the operation and close the import module.
- Regardless the choice of the type of import to do, error checking will first be conducted. If no critical errors are detected – the user will be required to view any warning messages and will then permitted to continue the import process as described previously.

3.1.8.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference

1.4.2.

3.1.9 Provider Inventory Data

3.1.9.1 Import file formats:

3.1.9.1.1 Database Files:

Source Table in VACMAN: provhas.dbf

Field Name	Type	Size
PIN	Char	6
IDATE	Date	8
VACCINE	Char	8
LOT	Char	15
POSITION	Numeric	2,0
DOSES	Numeric	8,0
CHGDATE	Date	8
USERID	Char	8
PROVHASID	Integer	4

Note: In Visual VACMAN 3.0 the PROVHASID field is the primary key for the prohas table and must not be altered in any way as this may prevent the record from matching.

3.1.9.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.9.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.9.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.9.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.9.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.9.2 Data Import procedures:

- The user will select Provider Inventory Data, the import option (archive, update or add), browse to the file name on the interface screen, and click the continue button.
- The system will check for duplicates by matching PIN number or PROVHASID when present. Data matching and handling of duplicates will be similar to the rules described in section 3.1.2.2 with the exception that there is no calculation to be done.
- Choosing the Cancel button will abort the operation and close the import module.
- Regardless the choice of the type of import to do, error checking will first be conducted. If no critical errors are detected – the user will be required to view any warning messages and will then permitted to continue the import process as described previously.

3.1.9.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.1.10 New Varicella Orders Import

3.1.10.1 Import file formats:

3.1.10.1.1 Database Files:

Source Table in VACMAN: neworder.dbf

Field Name	**Type**	**Size**
rectype	Char	1
pin	Char	6
orddate	Char	12
ordNum	Char	10
orddose	Numeric	8,0
ordPIN	Char	6
ordsize	Numeric	4,0
size	Numeric	4,0
vaccine	Char	8
maker	Char	8
cost	Numeric	9,4
contract	Numeric	15,0

fcode1	Char	1
fdose1	Numeric	8,0
note	Char	50
chgdate	Date	8
statepo	Char	32
userid	Char	8
ndcno	Char	16
exctax	Numeric	9,4
seekid	Numeric	9,0
xmitdate	Date	8

Note: There is no export module in VV3 designed to create the above table. This import module will NOT import any new orders other than varicella vaccine.

3.1.10.1.2 Comma Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.2.

3.1.10.1.3 Space Delimited Files:

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.3.

3.1.10.1.4 Microsoft Excel Spreadsheet files (.xls files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.4.

3.1.10.1.5 Lotus Spreadsheet files (.wk1 files)

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.5.

3.1.10.1.6 Other file formats

File definitions and format for this portion of the import module are identical to those described in section 3.1.2.1.6.

3.1.10.2 Data Import procedures:

- The user will select Import New Varicella Orders, the appropriate file type to import, browse to the file name on the interface screen, select the appropriate options for automatically assigning Fund Code and State PO number then click the continue button.
- **Note:** Excess order checking – an option in VACMAN 2.63 is automatically checked in Visual VACMAN 3.0 – the option to check or not check this value will

no longer be available in this module.

- The system will check the import file for duplicate records, and varicella orders which have already been transmitted (the record in the import file matches a record in the neworder table based on a match of PIN, order date, and vaccine. Records which are duplicates or which have already been transmitted will not be imported.
- If the number of records left is less than 1, the user is prompted with an appropriate notice that there are no records to import.
- Records which are ready to be imported will be checked for errors as discussed in section 3.1.10.2.1.
- On successful completion of error checking the system will append the records from the import table into the neworder table

3.1.10.2.1 Error checking criteria

- Error checking is handled by the business object in Visual VACMAN 3.0. Data standards in the import module will exactly mirror the standards enforced for keyboard entry. For exact specifications of all business rules please see Reference 1.4.2.

3.2 External Interface Requirements

3.2.1 User Interfaces – No new interfaces to be described in this document.

3.2.2 Hardware Interfaces – No new interfaces to be described in this document.

3.2.3 Software Interfaces – No new interfaces to be described in this document.

3.3 Performance Requirements

The product enhancements will conform to the same performance requirements as are currently in place for the existing product and Visual VACMAN 3.0 in general.

3.4 Hardware Limitations

There are no specific hardware limitations other than those already in place with the existing product and Visual VACMAN 3.0 as a whole.

3.5 Attributes

There are no special attributes to this product not already inherent in Visual VACMAN 3.0 and the current release as a whole.

3.6 Other Requirements

None.

This document is provided to list the exact error checking done on a field by field basis by the import module. The error checking listed is also being done during normal keyboard entry of records in the various tables listed.

The information in this document is organized based on the basic import type – i.e., Distribution data, Physician Data, etc. The information is then further broken down to list the database table affected, the exact field being validated, what information it contains, and a list of the various error conditions the import program tests for.

Distributions via Project Depot

Table affected: distrib.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Input	clsBusiness.ckMulDate()	1. Date cannot be blank 2. Date must be a valid date 3. Date cannot be more than 5 years prior to current date 4. Date cannot be more than 1 year from value in dateout field.
Dateout	clsBusiness.ckMulDate()	1. Date cannot be blank 2. Date must be a valid date
PIN	clsBusiness.ckMulPIN()	1. Value cannot be blank 2. Value cannot be '999999' 3. Value must match a PIN number in the 'Provider' table (see table definitions document for additional information about the provider table) 4. Provider referenced with PIN must be an active provider.
FromPIN	clsBusiness.ckMulPIN()	1. Value cannot be blank 2. If 'action' field = 8, value cannot be same as PIN field above. 3. Value cannot be same as a depot level PIN number 4. Value cannot be '999999' 5. Value must match a value in the 'Provider' table (see table definitions document for additional information about the provider table) 6. Provider referenced with this value must be an active provider.
Action	frmImport.distribimport()	1. Value must be 1
Vaccine	clsBusiness.ckDistVacc()	1. Value cannot be blank 2. Vaccine name must match a value in the vaccref table. (See table definitions document for detailed information on vaccref table)
Lot	clsBusiness.ckDistLot()	1. Value cannot be blank 2. Lot number must match a lot number in the STOCK table. (See table definitions document for more information on the STOCK table)

		3. Lot cannot be expired

Doses	clsBusiness.ckDistLot()
	clsBusiness.xsMain()
		1. Value cannot exceed value in ‘remain’ field of stock table for matching lot number
		2. Value cannot exceed providers maximum allowance

Units	clsProcFile.frvdistribution()	1. Units must be an even multiple of value of doses/size in the lots table
Multiple fields	clsBusiness.ckDistDup()	1. All records are checked for duplication

Distributions (other)

Table affected: distrib.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Input	clsBusiness.ckMulDate()	1. Date cannot be blank 2. Date must be a valid date 3. Date cannot be more than 5 years prior to current date 4. Date cannot be more than 1 year from value in dateout field.
Dateout	clsBusiness.ckMulDate()	1. Date cannot be blank 2. Date must be a valid date
PIN	clsBusiness.ckDoxPIN()	1. Value cannot be blank 2. Value cannot be ‘999999’ 3. Value must match a PIN number in the ‘Provider’ table (see table definitions document for additional information about the provider table) 4. Provider referenced with PIN must be an active provider.
FromPIN	clsBusiness.ckMulPIN()	1. Value cannot be blank 2. If ‘action’ field = 8, value cannot be same as PIN field above. 3. Value cannot be same as a depot level PIN number 4. Value cannot be ‘999999’ 5. Value must match a value in the ‘Provider’ table (see table definitions document for additional information about the provider table) 6. Provider referenced with this value must be an active provider.
Action	clsBusiness.ckDoxAction()	1. Value must be 2,3,4,5,6,7,8,9,11,12
Vaccine	clsBusiness.ckDistVacc()	1. Value cannot be blank

		2. Vaccine name must match a value in the vaccref table. (See table definitions document for detailed information on vaccref table)
Lot	clsBusiness.ckDoxLot()	1. Value cannot be blank 2. Lot number must match a lot number in the STOCK table. (See table definitions document for more information on the STOCK table) 3. Lot cannot be expired
Doses	clsBusiness.ckDoxDoses()	1. Value cannot exceed value in 'remain' field of stock table for matching lot number
Units	clsProcFile.frvdistribution()	1. Units must be an even multiple of value of doses/size in the lots table
Multiple fields	clsBusiness.ckDistDup()	1. All records are checked for duplication

Varicella Order Data

Table affected: stock.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Multiple	frmImport.varicellaImport()	1. Duplicate records are not allowed
Vaccine	frmImport.varicellaImport()	1. Value must be 'VAR'
Ndcno	frmImport.varicellaImport()	1. Insure that value is same as matching value for varicella in the vaccbulk table. For more information about the vaccbulk table see the table definitions document
Orddate	clsBusiness.ckMulOrdDt()	1. Date cannot be blank 2. Date must be a valid date 3. Date must be within 30 days of current date.
PIN	clsBusiness.ckVarPIN()	1. PIN cannot be blank 2. PIN must be in provider table (see the table definitions document for additional information about the provider table. 3. PIN must match an active provider 4. Contact, shipping and address information for matching record in provider table must be complete
PIN	clsBusiness.ckEligProv()	1. PIN must not match a public provider 2. PIN must match a varicella eligible provider
Fcode1	clsBusiness.ckVarFundCd()	1. Value cannot be blank
Statepo	clsBusiness.ckVarFundCd()	1. If value in fcode1 is 1 or 2 statepo field must be blank. 2. If value of fcode1 is other than 1 or 2, this value

		cannot be blank
Orddose Ndcno	clsBusiness.ckVarDose()	1. NDCno value must match a record in the vaccbulk table (for more information about the vaccbulk table see the table definitions document). 2. Total order may not exceed $25,000. 3. Orddose must be an even multiple of pkgsize in the vaccbulk table.
Rectype	frmimport.varicellaImport()	1. Value must be 2.
Ordpin	frmimport.varicellaImport()	1. Value must be the same as PIN field
Pkgsize	frmimport.varicellaImport()	1. Value must match the pkgsize field in the matching vaccbulk table record.
Maker	frmimport.varicellaImport()	1. Value must match the maker field in the matching vaccbulk table record.
Contract	frmimport.varicellaImport()	1. Value must match the contract field in the matching vaccbulk table record.
Cost	frmimport.varicellaImport()	1. Value must match the cost field in the matching vaccbulk table record.

Provider Data

Table affected: provider.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
PIN	clsBusiness.ckProvPIN()	1. Value cannot be empty 2. Value cannot be zero 3. Value cannot be ‘999999’ 4. Value cannot be ‘999998’ 5. Value cannot be duplicated 6. Value cannot already be in use in the Stock table.
linkPIN	clsBusiness.ckParPIN()	1. Sector field in provider table for record matching this value must be between 10 and 29, or must be 32.
Lastname	clsBusiness.ckMulName	1. Value cannot be empty
Sector	clsBusiness.ckProvSector()	1. Value cannot be 30, 31 or 33 2. Value must match a record in the sectoref table (see the table definitions document for additional information).
County	clsBusiness.ckProvCounty()	1. Value must match a record in the county table (see the table definitions document for additional information).
Subdiv	clsBusiness.ckProvSubD()	1. Value must match a record in the subdiv table (see the table definitions document for

		additional information).
Cat1	clsBusiness.ckProvCat()	1. Value must either be empty or must match a value in the cat1 field of the cat1 table (see the table definitions document for additional information).
Cat2	clsBusiness.ckProvCat()	1. Value must either be empty or must match a value in the cat2 field of the cat2 table (see the table definitions document for additional information).
Cat3	clsBusiness.ckProvCat()	1. Value must either be empty or must match a value in the cat3 field of the cat3 table (see the table definitions document for additional information).
Cat4	clsBusiness.ckProvCat()	1. Value must either be empty or must match a value in the cat4 field of the cat4 table (see the table definitions document for additional information).
Cat5	clsBusiness.ckProvCat()	1. Value must either be empty or must match a value in the cat5 field of the cat5 table (see the table definitions document for additional information).
Codes	clsBusiness.ckMulCode()	1. Value must be either 'Y' or 'N'
Active	clsBusiness.ckMulActive()	1. Value must be 'Y', 'N', 'S' or 'T' 2. If modifying the record and the old value was 'Y' being changed to 'N', 'S' or 'T', there cannot be an active record for ordering varicella in the neworder table (see the table definitions document for additional information). 3. If modifying the record and the value is not 'Y', the date value for dateXVFC field cannot be empty.
DateVFC	clsBusiness.ckProvVFCdt()	1. Value must be a valid date if used. 2. Value cannot exceed current date by more than 30 days.
Contact	clsBusiness.ckMuladdrship()	1. Value cannot be blank.
Address	clsBusiness.ckMuladdrship()	1. Value cannot be blank.
Phone	clsBusiness.ckMuladdrship()	1. Value cannot be blank. 2. Value must be properly formatted as: (999) 999-9999
City	clsBusiness.ckMuladdrship()	1. Value cannot be blank.
State	clsBusiness.ckMuladdrship()	1. Value cannot be blank.
Zipcode	clsBusiness.ckMuladdrship()	1. Value cannot be blank 2. Value SHOULD match an existing value in the zipcode table (see table definitions document for additional information).

<table>
<tr><td>Fax</td><td>clsBusiness.ckMuladdrship()</td><td>1. Value must be properly formatted if used as:
(999) 999-9999.</td></tr>
<tr><td>Modem</td><td>clsBusiness.ckMuladdrship()</td><td>1. Value must be properly formatted if used as:
2. (999) 999-9999.</td></tr>
<tr><td>Note</td><td>clsBusiness.ckProvNote()</td><td>1. Import module will record warning message if one of the following characters referenced is present in this field: !, @, #, $, %, ^, &, *, (,), +, |, \, /, {, }, [,], :, <, >, ?, '."
2. A warning message will be generated if the field is blank – should enter “NONE” if no value is available.</td></tr>
<tr><td>ML_address
ML_addres2
ML_City
ML_State
ML_ZipCode</td><td>clsBusiness.clMulMAddr</td><td>1. The mailing address information, if used is checked in a manner similar to that done in ckMuladdrship() method mentioned above, however this information is optional and will not generate a fatal error.</td></tr>
</table>

Enrollment Data

Table affected: enroll.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
PIN	clsBusiness.ckMulPIN()	1. Value cannot be blank 2. Value cannot be ‘999999’ 3. Value must match a PIN number in the ‘Provider’ table (see table definitions document for additional information about the provider table) 4. Provider referenced with PIN must be an active provider.

import.allkids1
import.vfckids1
import.g317kids1
import.statekids1
	clsBusiness.ckEnrolCnts()	1. Sum of vfckids, g317kids and statekids must be less than or equal to allkids field.

import.allkids2
import.vfckids2
import.g317kids2
import.statekids2
	clsBusiness.ckEnrolCnts()	2. Sum of vfckids, g317kids and statekids must be less than or equal to allkids field.

import.allkids3
import.vfckids3
import.g317kids3
import.statekids3
	clsBusiness.ckEnrolCnts()	3. Sum of vfckids, g317kids and statekids must be less than or equal to allkids field.

import.allkids4
import.vfckids4
import.g317kids4
import.statekids4
	clsBusiness.ckEnrolCnts()	4. Sum of vfckids, g317kids and statekids must be less than or equal to allkids field.

Physician Data

Table affected: physicn.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
PIN	clsBusiness.ckPhysPIN()	1. Value must match a PIN number in the 'Provider' table (see table definitions document for additional information about the provider table) 2. Matching provider record must have a sector field value between 10 and 29 or 32 3. Provider referenced with PIN must be an active provider. 4. Value cannot be blank.
Lastname	clsBusiness.ckMulLName()	1. Value cannot be blank.
Addtocount	frmImport.physImport()	1. Value must be 0 or 1 (equates to false or true)
Active	frmImport.physImport()	1. Value must be 'Y' or 'N'
Dateactive	frmImport.physImport()	1. Value must be a valid date. 2. Value cannot be blank.

Provider Inventory Data

Table affected: provhas.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Idate	clsBusiness.ckPHasIDt()	1. Date must be a valid date. 2. Date cannot be blank. 3. Date cannot be more than 5 years greater than or less than current date.
PIN	clsBusiness.ckMulPPin()	1. Value cannot be blank. 2. Sector field value in matching provider record must be between 10-29 or 32 3. There must be a matching, non-deleted record in the provider table. 4. Matching provider record must be active
Vaccine	clsBusiness.ckMulVacc()	1. Value cannot be blank. 2. Value must be present in vaccref table.
Lot	clsBusiness.ckPHasLot()	1. Must be a matching vaccine + lot number record in the stock table
Doses	frmImport.phasImport()	1. Value must be greater than zero.

Doses Administered Data

Table affected: dosesadm.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Adate	clsBusiness.ckDoAdmADt()	1. Date must be a valid date. 2. Date cannot be blank. 3. Date cannot be more than 5 years greater than current date.
PIN	clsBusiness.ckMulPPin()	1. Value cannot be blank. 2. Sector field value in matching provider record must be between 10-29 or 32 3. There must be a matching, non-deleted record in the provider table. 4. Matching provider record must be active
Vaccine	clsBusiness.ckMulVacc()	1. Value cannot be blank. 2. Value must be present in vaccref table.
Lot	clsBusiness.ckPHasLot()	2. Must be a matching vaccine + lot number record in the stock table
Doses	clsBusiness.ckDoAdmDose()	1. Value must be greater than zero. 2. Value may not exceed value in the maxno field of the vaccref table.

New Varicella Orders

Table affected: neworder.dbf		
Field	**Class/Method/Program Called**	**Validation Steps**
Xmitdate	frmImport.newOrdersImport()	1. Value must be empty.
Seekid	frmImport.newOrdersImport()	1. Value must be empty.
Orddose Fdose1	frmImport.newOrdersImport()	1. Value of orddose must be same as fdose1. If not the program will copy value of fdose1 into this field and will store a warning message.
Vaccine	frmImport.newOrdersImport()	1. Value must be ‘VAR’.
Ndcno	frmImport.newOrdersImport()	1. Value must match the ndcno field in the vaccbulk table.
Orddate	clsBusiness.ckOrdDt()	1. Must be a valid date. 2. Cannot be empty. 3. Must be less than current date – 30
PIN	clsBusiness.ckVarPIN()	1. PIN cannot be blank 2. PIN must be in provider table (see the table definitions document for additional information about the provider table.

		3. PIN must match an active provider 4. Contact, shipping and address information for matching record in provider table must be complete
PIN	clsBusiness.ckEligProv()	1. PIN must not match a public provider 2. PIN must match a varicella eligible provider
Fcode1	clsBusiness.ckVarFundCd()	1. Value cannot be blank
Orddose Ndcno	clsBusiness.ckVarDose()	1. NDCno value must match a record in the vaccbulk table (for more information about the vaccbulk table see the table definitions document). 2. Total order may not exceed $25,000. 3. Orddose must be an even multiple of pkgsize in the vaccbulk table.
Rectype	frmImport.newOrdersImport()	1. Value must be 2
Statepo	frmImport.newOrdersImport()	1. If fcode1 is 3, cannot be blank. 2. If fcode1 is 1 or 2 must be blank.
Ordpin	frmImport.newOrdersImport()	1. Value must be 'MERCK'
Size	frmImport.newOrdersImport()	1. Value must match the pkgsize field value in vaccbulk table.
Maker	frmImport.newOrdersImport()	1. Value must match the maker field value in vaccbulk table.
Contract	frmImport.newOrdersImport()	1. Value must match the contract field value in vaccbulk table.
Cost	frmImport.newOrdersImport()	1. Value must match the cost field value in vaccbulk table.